

# PATENT ABSTRACTS OF JAPAN

(11)Publication number : 07-334541  
 (43)Date of publication of application : 22.12.1995

(51)Int.CI. G06F 17/50

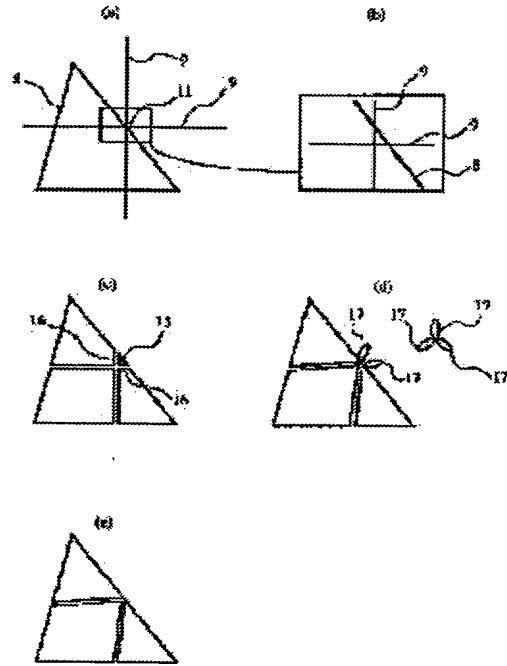
(21)Application number : 06-126080 (71)Applicant : HITACHI LTD  
 (22)Date of filing : 08.06.1994 (72)Inventor : YAMAMOTO YUSAKU  
 SAGAWA NOBUTOSHI  
 OKOCHI TOSHIO

## (54) ELEMENT GENERATION SYSTEM FOR NUMERICAL ANALYSIS

### (57)Abstract:

**PURPOSE:** To speed up a processing byunnecessitating an exceptional treatment which has been necessary conventionally.

**CONSTITUTION:** In the system dividing the element of an area by using a grid line which is previously given on the area, the exceptional treatment is not executed even if the angle of a division grid line 9 is just overlapped with an area boundary 8 to be divided 11. The area boundary is interpreted to pass through a point detached from the angle by infinitesimal, and division is executed normally. Thus, the side 16 of an infinitesimal length and an infinitesimal polygon 15, which are generated as the result, are removed after division. Thus, an element for numerical analysis is generated.



### LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

Copyright (C) 1998,2003 Japan Patent Office

(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平7-334541

(43)公開日 平成7年(1995)12月22日

(51)Int.Cl.<sup>6</sup>  
G 0 6 F 17/50

識別記号  
0834-5H

F I

技術表示箇所

G 0 6 F 15/ 60

4 5 0

審査請求 未請求 請求項の数3 O L (全9頁)

(21)出願番号 特願平6-126080

(22)出願日 平成6年(1994)6月8日

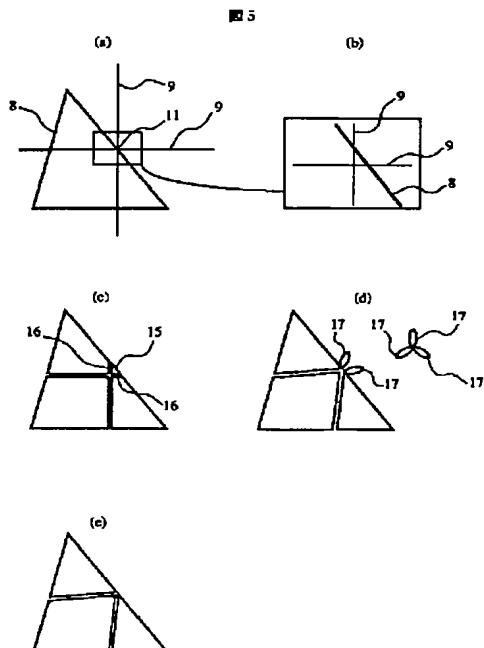
(71)出願人 000005108  
株式会社日立製作所  
東京都千代田区神田駿河台四丁目6番地  
(72)発明者 山本 有作  
東京都国分寺市東恋ヶ窪1丁目280番地  
株式会社日立製作所中央研究所内  
(72)発明者 佐川 暢俊  
東京都国分寺市東恋ヶ窪1丁目280番地  
株式会社日立製作所中央研究所内  
(72)発明者 大河内 俊夫  
東京都国分寺市東恋ヶ窪1丁目280番地  
株式会社日立製作所中央研究所内  
(74)代理人 弁理士 小川 勝男

(54)【発明の名称】 数値解析用要素作成システム

(57)【要約】

【構成】領域上に予め与えられた格子線を用いて領域の要素分割を行うシステムにおいて、分割用格子線9の角と分割すべき領域境界8とがちょうど重なった場合11でも、例外処理を行わず、領域境界が角から無限小だけ離れた点を通ると解釈して通常どおりに分割を行い、その結果生じる無限小長さの辺16や無限小の多角形15を分割後に取り除くことにより、数値解析用要素を生成する。

【効果】従来必要であった例外処理を不要にすることにより、処理を高速化できる。



## 【特許請求の範囲】

【請求項1】与えられた多面体領域または多角形領域を数値解析用の小領域に分割するシステムのうち、分割すべき領域を含む領域にあらかじめ与えられた規則的もしくは不規則的メッシュを用いて初期分割を行うシステムにおいて、

前記分割すべき領域の境界に属する頂点とメッシュに属する頂点もしくは辺もしくは面との間に重なりがある場合に、前記分割すべき領域の境界に属する頂点を前記メッシュに属する頂点から無限小だけずれた位置にあるとみなし、重なりのない場合と同様にして前記メッシュに属する辺もしくは面に対する頂点の内外判定を行い、その結果を用いて前記メッシュに属する辺もしくは面による領域の分割を行うことにより、重なりの有無、あるいは重なりの種類によって従来必要であった例外処理を用いることなく初期分割を終了し、前記無限小の辺を導入したことによって分割結果の小領域に生じる、長さが無限小の辺、面積が無限小の多角形、および体積が無限小の多面体を分割後に取り除くことを特徴とする数値解析用要素分割システム。

【請求項2】与えられた多面体領域または多角形領域を数値解析用の小領域に分割するシステムのうち、分割すべき領域と分割結果の小領域とを共に多角形領域とするシステムにおいて、

初期分割の際、前記多角形領域の境界に属する頂点と前記分割用メッシュに属する頂点もしくは辺との間に重なりがある場合に、前記多角形領域の境界に属する頂点を前記分割用メッシュに属する頂点から無限小だけずれた位置にあるとみなして前記重なりを取り除き、重なりのない場合と同様にして、メッシュの辺により前記多角形領域を前記辺の一方の側にある複数個の小多角形と前記辺の他方の側にある複数個の小多角形とに分割することにより、重なりを処理するために従来必要であった例外処理を用いることなく初期分割を終了し、

前記無限小の辺を導入したことによって初期分割の結果の小多角形に無限小長さの辺が生じた場合は、前記辺の両端点を重ね合わせて前記辺を除去することにより、前記多角形領域を有限長さの辺のみからなる小多角形に分割することを特徴とする数値解析用要素分割システム。

【請求項3】請求項2において、あらかじめ与える前記分割用メッシュとして、場所により格子の疎密を変化させた直交格子を用いる数値解析用要素分割システム。

## 【発明の詳細な説明】

## 【0001】

【産業上の利用分野】本発明は数値解析モデルの作成方法に係り、特に、有限要素法等を行う場合に解析領域を分割して数値解析用の有限要素を自動生成する要素生成システムに関する。

## 【0002】

【従来の技術】偏微分方程式により記述される物理現象の解析方法として、有限要素法による数値シミュレーションが広く用いられている。この方法は解析領域を多数の小領域（有限要素）に分割し、各小領域で偏微分方程式を離散化することにより、偏微分方程式を各要素上の節点での物理量についての連立一次方程式に帰着させて解を求めるものである。節点数は通常きわめて多くなるため、シミュレーションを効率的に行うためには、解析領域の形を与えるだけで自動的に領域分割を行い、有限要素を生成するシステムが望まれる。

【0003】解析領域の形を与えて有限要素を自動生成するシステムは、一般に、図1のように解析領域を入力するための入力装置、入力された領域上に要素を生成するための演算装置、生成した有限要素を出力するための出力装置からなる。このうち演算装置で要素を作成するために用いられる方法は、正規領域に張られた格子から写像関数によって解析領域上の格子を作る方法（たとえば J. F. トンプソン(Thompson), 他著、"ニューメリカル グリッド ジェネレーション(Numerical Grid Generation)" , North-Holland, 1985 参照）、デローネ三角形分割を利用する方法（たとえば S. リベイ(Rebay) 著：“Efficient Unstructured Mesh Generation by Means of Delaunay Triangulation and Bower-Watson Algorithm”，ジャーナル オブ コンピュティショナル フィジクス (Journal of Computational Physics), Vol. 106, pp. 125-138(1993) 参照）、分割すべき領域を含む領域にあらかじめ与えられた規則的もしくは不規則的格子を用いて分割を行う方法（たとえば P. L. ベーマン(Baehmann) , 他著 “Robust, Geometrically Based Automatic two-Dimensional Mesh Generation, インタナショナル ジャーナル フォー ニューメリカル メソズ イン エンジニアリング (International Journal for Numerical Methods in Engineering), Vol. 24, pp. 1043-1078(1987) 参照）などが知られている。この三者のうち最後にあげた方法は、複雑な領域形状の場合でも、大きさのばらつき、極端なつぶれなどのない高品質な要素を生成することが比較的容易であり、かつ要素生成に要する時間が短いため、多くの利用例がある。2次元の解析領域を要素分割する場合、この方法の一種である四分木法（上記最後の文献参照）が広く利用されている。以下、四分木法の概略を示す。四分木法は次の5ステップから構成される。

## 【0004】(1) 四分木の生成

解析領域の境界を多角形と仮定する。四分木法では、まず解析領域を含む平面上に基本となる直交格子を引く。これにより平面は正方形に分割されるが、細かい解析要素が必要な領域ではこの正方形を4分割し、さらに細かい解析要素が必要な領域ではさらに4分割する、ということを繰り返す（図2(a)）。この再帰的な4分割によって最終的にできた各正方形を、以後では小正方形と呼

ぶ。また、小正方形の角、解析領域の境界多角形の頂点、格子線と領域境界とが交差してできる交点を、以後ではそれ単に角、頂点、交点と呼ぶことにする。さらに、再帰的な4分割によってできる小正方形の木構造を四分木と呼ぶ(図2(b))。

【0005】(2)格子線による領域の分割

上で作った格子線を粗いものから順に取り、境界多角形が該格子線と交差するか否かを判定し、交差する場合には該格子線によって該多角形を複数個の小多角形に分割する(図3)。この手続きをすべての格子線について繰り返すことにより、対象領域の小多角形への分割を得る(図2(c))。

【0006】なお、本処理における交差判定および分割では、分割に用いる格子線に対して境界多角形の各頂点が上側(右側)にあるか下側(左側)にあるかを判定し、隣り合う頂点が反対側にあるときに交差があるとみなして分割を行っている。しかしこの方式では、頂点がちょうど格子線と重なっているときは扱えず、例外処理が必要となる。「発明が解決しようとする課題」の項で後述するように、考慮しなくてはならない重なり方は4種類あるため、例外処理によって重なりに対処する方式では処理が複雑となり、実行速度が低下するという問題がある。

【0007】(3)点の重ね合わせ

上記分割の結果、きわめて短い辺を持つ小多角形、あるいは細長い小多角形ができることがある。これは後の要素生成において形の悪い要素を作る原因になるので、その前に取り除いておかなくてはならない。そのための方法は、近い位置にある二つの点を重ねること、線分の近くにある点を線分上に乗せることの二つがある。これらの操作を点の重ね合わせと呼び、実用になる有限要素を作るには不可欠の段階である。図2(c)の領域分割に対し、点→点の重ね合わせおよび点→境界辺の重ね合わせを行うことにより、極めて短い辺および細長い多角形を除去した結果を図2(d)に示す。

【0008】(4)要素の生成

各小多角形を適当に分割して三角形要素を作る(図2(e))。

【0009】(5)スムージング

要素の節点位置を適当に移動させることにより、要素のゆがみをなくす(図2(f))。

【0010】以上で、対象領域の三角形要素への分割が完了する。これが従来の四分木法である。

【0011】

【発明が解決しようとする課題】四分木法では、(2)の分割が全体の実行時間の大部分を占める。この分割は有限要素を作るための領域の初期分割であり、与えられた格子により、多角形領域を有限長さの辺のみを持つ小多角形に分割することを目的とする。

【0012】本分割処理では、上述のように、分割に用

いる格子線に対して境界多角形の各頂点が上側(右側)にあるか下側(左側)にあるかを判定し、隣り合う頂点が反対側にあるときに交差があるとみなして分割を行う。しかし、この方式では、頂点がちょうど格子線と重なっているときは扱えず、例外処理が必要となる。このように、格子と境界多角形との間に重なりがあり、分割のために例外処理が必要となる状況を、ここでは退化と呼ぶ。

【0013】四分木法の分割処理における退化は、(1)境界辺が格子の角を通る場合、(2)多角形の頂点が格子の角と一致する場合、(3)多角形の頂点が格子線上にある場合、(4)多角形の辺と格子線とが重なる場合の四通りが考えられる(図4)。そこで従来の方法では、各交差判定において、退化が存在するか否か、存在する場合には上記4種類のうちのどれかを判定し、それぞれの場合に応じた分割を行っていた。しかし、その結果、各交差判定において複雑な場合分けと例外処理が必要となり、実行速度が低下するという問題があった。

【0014】本発明の目的は、退化を統一的な手法で扱って例外処理をなくすことにより、例外処理によって生じていたプログラムの速度低下の問題を解決しようとするものである。

【0015】

【課題を解決するための手段】上記目的を達成するため、本発明による有限要素生成システムでは無限小の長さという新概念を導入した。退化が生じるのは、入力の境界多角形に属する点あるいは線分が格子に属する点あるいは線分と重なるからである。そこで本発明では、重なりが生じたとき、前者が後者よりも無限小だけ上あるいは右にあるとみなし、重なっていない場合と同様の処理を行う。これにより、今まで必要であった「重なり」という概念はなくなり、交差判定の結果は交差するか否かの2通りで記述できることになる。これに伴い、今まで必要であったすべての例外処理は不要となる。

【0016】ただし、ここで導入した無限小長さを放置すると、最後の要素生成の段階において、無限小長さの辺を持った要素ができてしまう可能性がある。そこで、もともとの四分木法にあった点の重ね合わせの手続きを用いて、無限小長さを自動的に解消する。これにより、退化の存在によって生じた無限小長さは、最終的に生成された要素には現れないことが保証される。

【0017】

【作用】1ヶ所だけに退化(境界線と格子の角との重なり)がある場合を例にとり、例外処理なしに退化が解消される過程を説明する。

【0018】まず、図5(a)が入力図形と格子との位置関係である。領域境界として入力された三角形の一辺と格子の角とが重なっているが、本発明の方法ではこの場合、入力図形の辺が角の無限小だけ右上を通っていると解釈する。この部分を拡大したのが図5(b)である。格

子線により入力図形を分割したのが図5(c)であるが、例外処理は行われず、そのかわりに右上部分に無限小の三角形ができる。また、左上、右下の図形にもそれぞれ無限小長さの辺が生じる。

【0019】次に、点の重ね合わせを行ったものが図5(d)である。点の重ね合わせにより、無限小長さの辺の両端点は同一の点に重ねられる。なお、この点の重ね合わせの手続きは、従来の四分木法で有限の微小長さの線分を解消するのに用いていたものをそのまま使えばよい。点の重ね合わせにより、左上と右下の図形は三角形に無限小の自己閉路が付いたものになる。また、右上の無限小の三角形は、一つの点から三つの自己閉路が出た形になる。最後にこれらの自己閉路を除去すれば、入力図形は三つの小図形に分割される。これを図5(e)に示す。自己閉路の除去にも、従来の四分木法で用いていたものをそのまま使えばよい。

【0020】以上により、退化のある場合でも例外処理なしに入力図形の分割を正しく行うことができた。

【0021】

【実施例】以下、本発明の原理および実施例を、図面により詳細に説明する。ここで実施例として挙げるのは、与えられた2次元領域を有限要素法用の三角形要素に分割するシステムである。システムは領域形状を入力するための入力装置、領域を要素に分割するための演算装置、分割した要素を出力するための出力装置から構成される。

【0022】演算装置における領域分割処理は、図6に示すステップから構成される。本実施例では、まず入力装置から領域形状データおよび要素分割の疎密制御データを入力し(図6の22)、後者に基づいて領域平面上に疎密を持った直交格子を生成し(図6の23)、この格子によって領域の境界多角形を分割して行くことにより、最終的に領域の三角形要素への分割を得る。

【0023】このうち直交格子生成後のステップは、前半部(図6の25, 26, 27, 28, 29)と後半部(図6の31, 32, 33, 34)とに大きく分けられる。

【0024】前半部における基本操作は、1本の格子線Lをとり、小多角形リスト(図6の30)の中から1個の小多角形Cを取り出し、LとCとが交わるならばLによってCを分割し、分割によって得られた複数の小多角形を再び小多角形リストに戻すことである。小多角形リストに境界多角形のみが入っている状態(図6の24)から出発し、すべての格子線、およびその格子線に対して小多角形リスト中のすべての小多角形に対してこの基本操作を繰り返すことにより(図6の25, 26)、前半部が終了する。

【0025】一方、後半部での基本操作は、小多角形リストの中から1個の小多角形Cを取り出し、その形状を後述のように修正した後、これを三角形要素に分割する

ことである。小多角形リスト中のすべての小多角形に対してこの基本操作を繰り返すことにより(図6の31)、後半部が終了する。最後にスムージング操作によって三角形要素のゆがみを取り除き、作成した要素を出力装置に出力する。

【0026】本発明における最も大きな特徴は、上述の前半部および後半部の基本操作における退化の取り扱い方法にあるので、以下、最初に前半部の基本操作である図6の27, 28, 29のそれぞれを(1), (2), (3)として詳しく述べ、次に後半部の基本操作である図6の32, 33, 34のそれぞれを(4), (5), (6)として述べる。

【0027】(1) 交点計算(図6の27)

ここでは小多角形リストから取り出した小多角形Cと格子線Lとの交点を計算し、求めた交点を多角形Cの辺に挿入する。それには多角形Cの隣り合う頂点v<sub>i</sub>, v<sub>j</sub>をとり、v<sub>i</sub>とv<sub>j</sub>が直線Lの反対側にあつたら、交点uを計算して辺v<sub>i</sub>v<sub>j</sub>上に挿入すればよい。いま、多角形Cの辺には反時計回りに向きがついているとする。このときCの境界辺と格子線Lの交点の符号を、辺が格子線Lを下から上へ横切るとき+、上から下へ横切るとき-と定義すると、こうして得られる交点には明らかに+と-のものが同数個ある(図7(a))。

【0028】以上ではCの頂点が格子線L上にないことを暗黙のうちに仮定していたが、頂点が直線L上に乗っている場合には、直線Lの上側にあるものと定義し、通常の場合と同様にして必要ならば交点を求める。たとえば、図7(b)およびその拡大図に示すように、多角形Cが格子線Lに上側から接する場合には交点がないものとみなし、下側から接する場合には2個の交点があるとみなす。これにより退化を特別扱いせずにすみ、また、交点の符号の数についての上述の性質は退化がある場合にも成り立つことになる。このとき頂点と交点とを結ぶ辺は無限小長さとなるが、以後での取扱いは有限長さの辺の場合とほとんど変わらない。

【0029】(2) 交点の整列(図6の28)

上で求めた交点は、格子線L上では左から-, +, -,+のように1個おきに向きが逆のものが並んでいるはずである。しかし実際には数値誤差のため、こうならないこともある。特にここではCの頂点が格子線L上にある場合でも交点を計算しているため、順序の逆転が起こってしまう可能性が大きい。順序が逆転すると、次のステップの多角形Cの分割がうまく行かず、多角形Cの内と外とが反転してしまうことも起こりうる。

【0030】そこで、格子線L上で交点の並べ替えを行い、交点が左から正しく-+-+と並ぶようにする(図8)。並べ替えは、計算された交点を左から順に見てゆき、符号がおかしいところがあつたら隣と入れ換えることによって行う。この手続きによれば、計算誤差により交点順序が逆転した場合でも、正しい順序が復元でき

る。

【0031】(3) 多角形の分割(図6の29)

上で得た交点を使って、多角形Cを格子線によって複数個の多角形に分割する。(2)のステップにより交点は順序正しく並んでいるはずだから、分割は問題なく行える(図9(a))。なお、退化のある場合は分割により無限小長さの辺を持つ多角形が生じるが、これはそのままにしておく(図9(b))。Cの分割によってできた複数の小多角形は、無限小長さの辺を持つものも含め、小多角形リスト(図6の30)に追加する。

【0032】小多角形リストに領域境界の多角形のみが入っている状態から出発し、すべての格子線L、および格子線について、小多角形リスト中のすべての小多角形Cに対して以上の(1)、(2)、(3)の手続きを適用することにより、境界多角形をもつとも細かい小多角形にまで分割できる。退化がある場合には無限小長さの辺を持つ小多角形が生じるが、特別な処理は不要である。これにより、前半部の操作が終了する。次に、以上の操作により作成された小多角形リスト中のすべての小多角形Cに対し、次の(4)、(5)、(6)の手続きを適用する。

【0033】(4) 点の重ね合わせ(図6の32)

小多角形リストから1個の小多角形Cを取り、点の重ね合わせという操作を行う。これは、きわめて短い辺を持つ小多角形、細長い小多角形などを取り除くための前処理であり、小多角形Cを構成する点のうち、近い距離にある二つの点を重ねること、および小多角形の辺に近い点を辺上に乗せることの二つからなる。以下、このそれについて説明する。点の重ね合わせを行うことにより、退化のために生じた無限小長さの辺も取り除くことができる。

【0034】(a) 点→点の重ね合わせ

小多角形Cの点には、もとの領域境界の頂点、境界と格子との交差によってできた交点、格子どうしが交わる角の3種類がある。ここでは点と点との距離がd以下のときに二つの点を重ねるが、このことによって領域境界の形状が変わってはならない。そこで、各点の移動に関しては次のような制限がつく。

【0035】(1) 頂点は動かしてはならない。

【0036】(2) 交点は、それが乗っている境界辺の上ののみを動くことができる。

【0037】(3) 角はどこへでも動ける。

【0038】これらより、重ね方に関しては次の規則ができる。

【0039】(1) 頂点は他の点に重ねてはならない。

【0040】(2) 交点は、同じ境界辺上にある他の頂点、交点に重ねることができる。

【0041】(「退化」という概念がないことより、境界辺上に角が乗ることはないことに注意。)

(3) 角はどの点にでも重ねることができる。

【0042】これらの制限を考慮した上で、重ねられる 50

点をすべて重ねる。なお、点の重ね合わせを行ったということは、重ねる点のデータ構造に、それが重なった相手の点の種類と点番号とを記入し、以後その点を重なった相手の点と同一視することにより表現する。小多角形C自体のデータ構造(構成する点の数など)は変更しない。したがって点→点の重ね合わせにより、小多角形Cには自己閉路、あるいはくびれ、並列枝などができることになる。

【0043】点の重ね合わせの例を図10に示す。○印の二つの点を重ね合わせることにより、自己閉路が生じる。

【0044】(b) 点→辺の重ね合わせ

小多角形Cの辺には、もともと領域の境界辺だったものと格子辺だったものの2種類がある。これと点の種類とを考え合わせると、可能な重ね方は次の2通りになる。

【0045】(1) 角を境界辺の上に乗せる(図11(a))。

【0046】(2) 頂点を格子辺の上に乗せる(図11(b))。

【0047】このうち(2)では、頂点の位置を移動することはできないので、実際には格子辺の方を曲げて頂点を通るようにする。これらの制限のもとで、辺から距離d以内にある点をすべて辺に重ねる。

【0048】点→辺の重ね合わせを行ったということは、小多角形Cのデータ構造において辺上に新たな点を設け、重ねる点と同じ点番号、点の種類をその点に与えることによって表現する。その他の小多角形Cのデータ構造は変更しない。したがって点→辺の重ね合わせにより、小多角形Cにはくびれあるいは並列枝ができることがある。

【0049】角→境界辺の重ね合わせを行った例を図11(a)に示す。右上の小三角形の○印の点を対辺に重ね合わせることにより、くびれと並列枝のある小多角形が生じる。また、頂点→格子辺の重ね合わせを行った例を図11(b)に示す。右側の小多角形の○印の頂点を向かいの格子辺に重ねることにより、くびれのある小多角形が生じる。

【0050】以上が点の重ね合わせの概略であるが、これをそのまま行っただけでは生成される要素の品質は必ずしも改善されない。実際、点→辺の重ね合わせを行ったために辺が二つに分割され、かえって短い辺ができることも起こりうる。そこで、点の重ね合わせによって新たに生じる辺の長さを前もって評価し、前よりも短い辺が生じないという条件のもとで、点の重ね合わせを実行する。

【0051】(5) 小多角形の簡約(図6の33)

点の重ね合わせの結果、小多角形Cは自己閉路、くびれ、あるいは並列枝などを含むようになる。したがって要素を生成するためには、これらのものを取り除く必要がある。これを行うのが簡約である。自己閉路、並列枝

などの発見は小多角形の点リストの番号だけを見て行えるので、簡約は完全に組合せ論的な操作であり、数値計算を必要としない。簡約の結果、小多角形Cはさらにいくつのか多角形に分割される。また、無限に小さい小多角形、完全につぶれた小多角形などは消滅する(図1-2)。

【0052】以上の(4)と(5)の操作により、退化は(隅に意識されることなく)完全に取り除かれる。また、これらは退化のない場合でも、形の良い要素を作るには不可欠の段階であるから、退化を考慮したために特に処理が複雑になったわけではない。

#### 【0053】(6) 三角形への分割(図6の34)

これは小多角形をさらに分割し、最終的な要素を作る段階である。この段階では特に退化の問題はないが、なるべく形の良い要素ができるような分割を行う必要がある。

【0054】以上の(4), (5), (6)の操作を小多角形リスト中のすべての小多角形に対して行うことにより、後半部が終了する。この後、次のスムージング操作を行うことにより、要素生成が終了する。

#### 【0055】(7) スムージング

最後に、点の位置を移動させることによって要素のゆがみを取り除く。動かすのは角および交点であり、角は領域内、交点は境界線上を移動させる。

【0056】以上が本発明を利用した2次元有限要素生成方法の基本的な部分であるが、これを使って実際にプログラムを作るには、次のような工夫が必要となる。

#### 【0057】(1) 安定な交点計算法

#### (2) 2段階の点の重ね合わせ

以下ではこれらについて説明する。

#### 【0058】(1) 安定な交点計算法

上のアルゴリズムでは多角形と直線との交点がいつでも計算できることを前提としているが、入力データによつては、多角形の辺と格子の辺とがほとんど平行になる場合も起こりうる。このようなとき、2直線の交点を求める通常の公式をそのまま適用したのでは、たちまちオーバーフローが生じてプログラムが異常終了してしまう。従来の方法ならこののような場合を退化として特別扱いすることもできたが、本アルゴリズムでは退化を特別扱いしないので、どんな場合でも1個の交点を計算して返さなくてはならない。

【0059】そこで、通常の公式を用いることをやめ、垂線を利用した安定な計算法を採用した。直線を1、線分(多角形の辺)の端点をp1, p2とするとき、これは次のようになる。

【0060】(1) p1, p2からそれぞれ1におろした垂線の足t1, t2を求める。

(2) 線分p1t1, p2t2の両方が微少量εよりも短いならば、(両端点が直線上に乗っていると判断して)t1を交点として採用する。

(2) そうでないならば、線分t1t2をp1t1:p2t2に分ける点を交点として採用する。

【0061】これによれば、線分の両端点が直線に関して反対側にありさえすれば、いかなる入力に対してもただ一つ交点が求められることが保証され、また交点のx, y座標が両端点のx, y座標の中間にあることが保証される。

#### 【0062】(2) 2段階の点の重ね合わせ

点の重ね合わせの過程では、無限小長さの辺と有限長さの短い辺の両方を取り除かなくてはならない。ところが、これを一度にやろうとすると、点の重ね合わせを行う順番によって、ある点が無限小だけ離れた点に重ねられず、有限の距離だけ離れた点に先に重ねられてしまうことが起こりうる。すると小多角形のゆがみが大きくなり、結果としてできる要素の形に悪影響が出る。この例を図13(a)に示す。図の一番左側の図形でεと書いてあるのが無限小長さの線分である。この図形においてA, B, Cと書いてある点がそれぞれ同じ記号の点に重ね合わせられた場合、点Cは同じ小多角形上の無限小だけ離れた点とは重ねられずに有限の距離だけ離れた点と重ねられるため、図形は細長く引き伸ばされ、結果として一番右の図のような細長い要素が生成される。

【0063】この問題を解決するため、まず基準となる距離を十分小さく(たとえば小正方形の辺の長さの1/100分の1程度)おさえて点の重ね合わせを行い、次に距離を大きく(たとえば辺の長さの4分の1程度)してもう一度点の重ね合わせを行う。このように2段階に分けることにより、無限小長さの辺に対して優先的に点の重ね合わせが行われることが期待できる。この方法による重ね合わせの例を図13(b)に示す。無限小だけ離れた点が最初に重ね合わされるため、点A, B, Cは図中の同じ記号の点と重なり、形の良い要素が生成される。

【0064】退化によって生じる場合分けの除去によるプログラムの高速化という本発明の基本的な特徴に加えて、以上の二つの工夫を採用することにより、生成される要素の形の改善、またいかなる入力に対してもオーバーフローによってプログラムが異常終了することのない頑健性といった特徴を持つ要素生成システムを構成することができる。

【0065】なお、本実施例においてはすべて2次元の領域の場合について説明したが、3次元の場合でも、八分木法として知られている要素生成方法を用いれば、本発明の退化処理方法はそのまま適用できる。また、本実施例では三角形要素を生成する場合について説明したが、本発明の方法は、これ以外の要素形についても全く同様に適用できる。

#### 【0066】

【発明の効果】本発明によれば、四分木法等を利用した領域分割システムにおいて、従来必要であった数多くの場合分けを設けることなく、退化のある場合を統一的に

11

扱うことができる。その結果、プログラムにおいて複雑な処理が不要となり、実行速度が向上する。

## 【図面の簡単な説明】

【図1】四分木法による有限要素生成システムの全体を示すブロック図。

【図2】従来の四分木法による有限要素生成方法の各ステップを示す説明図。

【図3】従来の四分木法において、与えられた領域多角形を格子線により分割して複数の小多角形を作る部分の説明図。

【図4】四分木法において起こりうる4種類の退化の説明図。

【図5】本発明の方法で退化が例外処理によらず自動的に解消されることを示す説明図。

【図6】本発明を四分木法による有限要素生成システムに適用した場合の手続き全体を示すフローチャート。

10

【図7】図5の手続きのうち交点計算を行う部分の説明図。

【図8】交点順序の逆転を検出し、補正を行う部分の説明図。

【図9】退化のない場合とある場合についての多角形の分割を行う部分の説明図。

【図10】点→点の重ね合わせの部分の説明図。

【図11】点→辺の重ね合わせの部分の説明図。

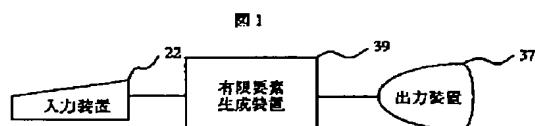
【図12】小多角形の簡約を行う部分の説明図。

【図13】2段階の点の重ね合わせの効果の説明図。

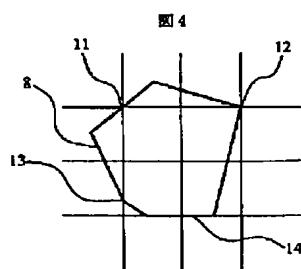
## 【符号の説明】

8…入力多角形の境界、9…格子線、11…境界辺が格子の角を通ることによって生じる退化、15…退化により生じる無限小の小多角形、16…無限小長さの辺、17…点の重ね合わせによってできる自己閉路。

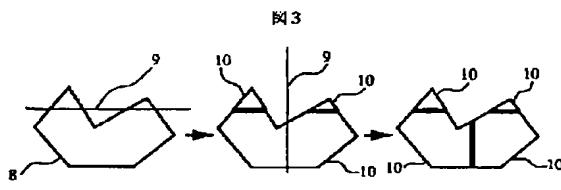
【図1】



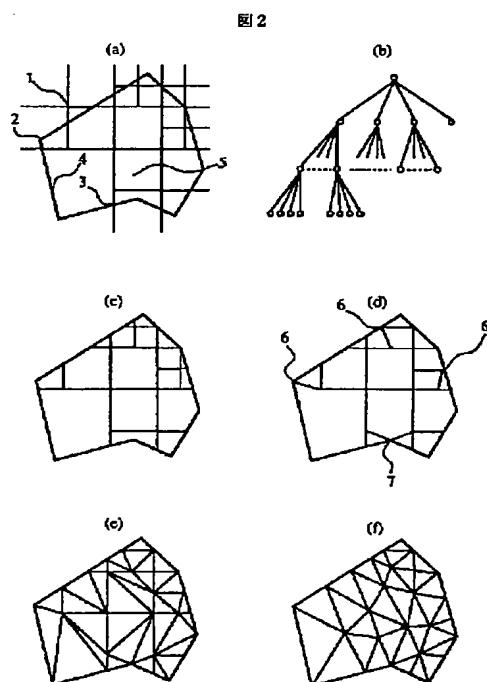
【図4】



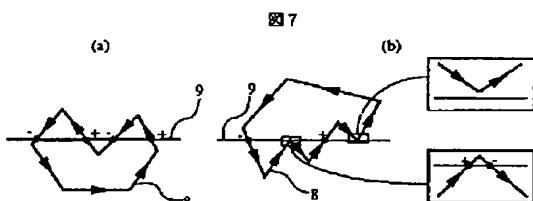
【図3】



【図2】

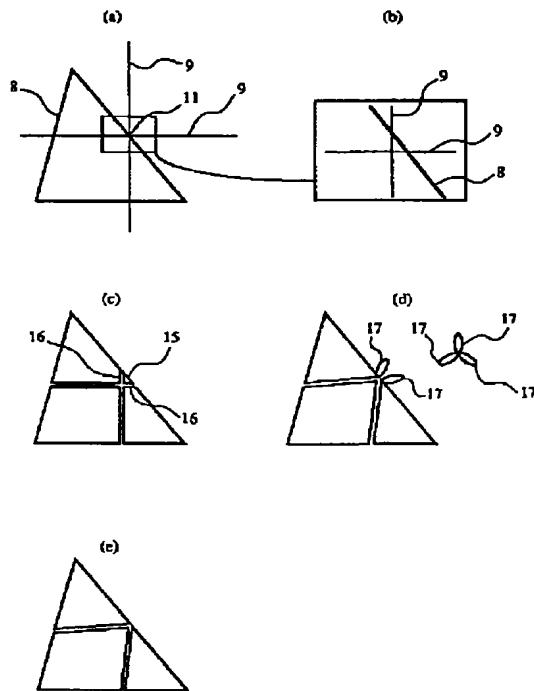


【図7】



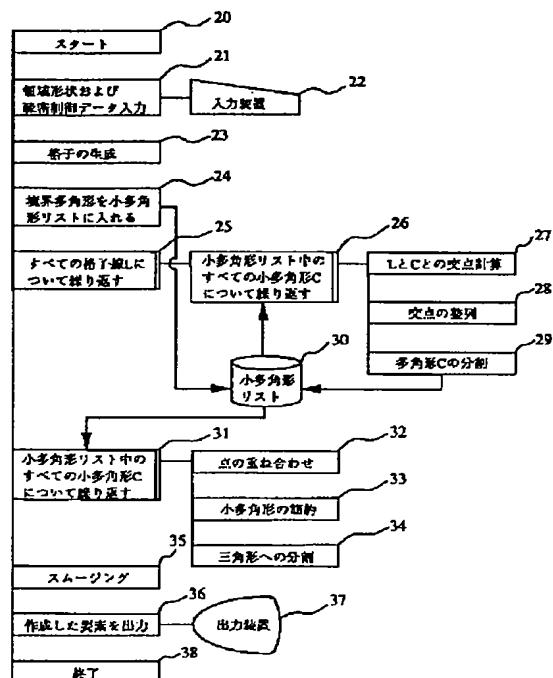
【図5】

図5



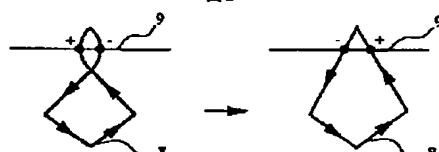
【図6】

図6



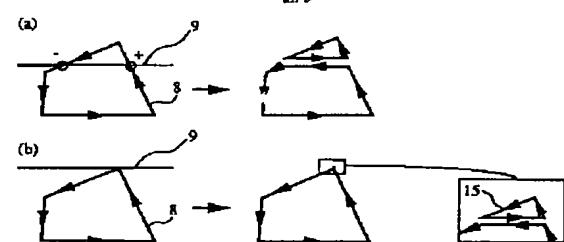
【図8】

図8



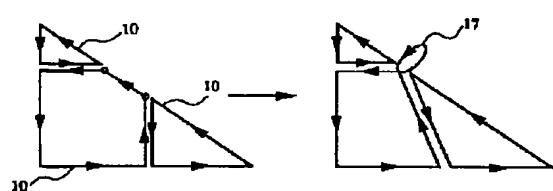
【図9】

図9



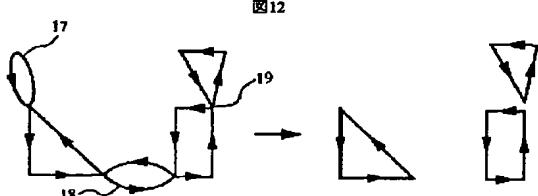
【図10】

図10



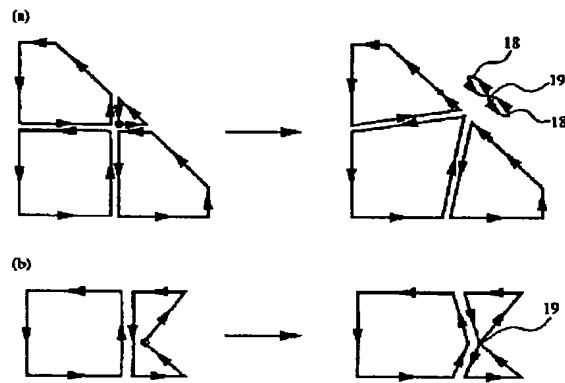
【図12】

図12



【図11】

図11



【図13】

図13

